DOCUMENT RESUME

ED 331 464 IR 013 761

AUTHOR Caldwell, Eveline M.
TITLE The Use of an Expert System in Human Resource Management.
PUB DATE Dec 88
NOTE 9p.; Paper presented at the Biennial Convention of South East Asia Regional Computer Confederation (7th, New Delhi, India, November 28-December 1, 1988). In "Modern Trends in Information Technology" P.V.S. Rao and P. Sadanandov, editors, Tata McGraw-Hill Publishing Company, Ltd., p. 85-92.
PUB TYPE Viewpoints (Opinion/Position Papers, Essays, etc.) (120) -- Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Software Evaluation; Evaluation; *Expert Systems; *Holistic Approach; Microcomputers; *Personnel Data; *Personnel Management; *Skill Analysis; *Skill Development; Task Analysis

ABSTRACT

The complexity of the optimization algorithm for staff utilization in any medium to large organization can cause major problems in meeting schedules while still keeping all of the work force in full employment, working at a level appropriate to their experience, and also giving staff an opportunity to develop new skills and experiences that will both enrich the company and also significantly contribute to the individual's job satisfaction. A prototype expert system has been developed which provides advice both on the staff suitable for a particular task and the nature of tasks suitable for a particular member of staff, while taking a holistic view of the work the organization seeks to undertake within a given time frame. The resources to be allocated were based on those of an academic computing department modelling the current staffing, teaching, and research commitments. The prototype running on an IBM PC XT was successful in achieving its original aim of ameliorating the difficulties encountered in human resource allocation in a real life situation. The major difficulties encountered were in the area of knowledge engineering and fact finding rather than design and implementation, although a production system would require a faster machine if it were to produce results in an acceptable response time. (2 figures and 10 references) (Author/EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED331464

MODERN TRENDS IN INFORMATION TECHNOLOGY
PVS Rao, P Sadanandan (Editors)
Tata McGraw-Hill Publishing Company Ltd
©CSI, 1988

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Eveline M.Caldwell
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The Use of an Expert System in Human Resource Management

Eveline M. Caldwell
Hong Kong Polytechnic, Hong Kong

Abstract

The complexity of the optimization algorithm for staff utilization in any medium to large organisation can cause major problems in meeting schedules while still keeping all of the work force in full employment, working at a level appropriate to their experience and also giving staff an opportunity to develop new skills and experiences which will both enrich the company and also significantly contribute to the individual's job satisfaction. This paper describes the development and evaluation of a prototype expert system which provides advice both on staff suitable for a particular task and the nature of tasks suitable for a particular member of staff while taking a holistic view of the work the organisation seeks to undertake within a given timeframe. The resources to be allocated were based on those of an academic computing department modelling the current staffing, teaching and research commitments. The prototype running on an IBM PC XT was successful in achieving its original aim of ameliorating the difficulties in human resource allocation in a real life situation. The major difficulties encountered were in the area of knowledge engineering and fact finding rather than design and implementation although a production system would require a faster machine if it were to produce results in an acceptable response time.

1. Introduction

A skilled, highly trained and experienced workforce is often an organisation's major expense and greatest asset. The task of optimising the match between the needs of the organisation and the wishes of the individual is extremely important to the smooth running of the organisation and essential to the achievement of strategic goals. Within any organisation assigning the most suitable person to a particular job is often a complex task. Consideration should be given to including the matching of the individual's specialism, experience, and interests as well as the overall optimisation of the human resources in the organisation. Under some circumstances these may create a conflict of interests between the aspiration of an individual and of the employer, and between individuals. This is particularly complex in an organisation which employs a number of similarly qualified professional staff who nevertheless, have a wide range of personal experience and aspirations. When new work has to be done by a team of professionals it is not unusual that many of them will have similar skills and common interests. However some of them will be already fully committed to existing work while others perhaps have some spare time. Existing work has deadlines to meet and the new job has therefore to be allocated to staff

IR 013 761

BEST COPY AVAILABLE

- who are capable of doing it,
- who have the time to do it,
- who can still meet existing deadlines for other work,
- who are interested in the nature of the work involved in the new task.

The Department of Computing Studies employs 57 computer professionals providing computer education for a Polytechnic of 26,000 students. It was becoming increasingly difficult to keep track of exactly who had done which course, who knew certain subject areas, who wanted to develop into a new area and so on. We decided that this was a job with which a computer could help . A conventional programming language could have been used but this would have imposed severe limitations and made the system very inflexible. It was therefore decided to develop a prototype expert system for evaluation.

2. Scope of the Project

There were three main parts to this project. Firstly came knowledge engineering - to build the knowledge base, extract decision criteria, rules, and heuristics from domain experts, and formalize them into key concepts and relations. Secondly we had to select a suitable combination of hardware and software for the implementation of the expert system. This was primarily based on particular characteristics of the knowledge conceptualised and available resources. Thirdly the implementation concerned mapping key concepts and relations into the formal representation suggested by the selected software. This third phase consisted of many iterations to increasingly conform to the domain expert's expected performance criteria. Our main goal was to evaluate the use of an expert system applied to a human resource allocation problem in a real-life situation.

3. The Problem Domain

Customarily we assigned staff to tasks on the basis of their experience, specialism and availability and of course the needs of the job. Individual preferences were also considered as far as possible with self-development of staff as an important goal. However the rules underlying the decision process were often unstructured and implicitly applied by the domain expert simultaneously. This is because many of the decision criteria were impossible to quantify. Also, different considerations are interrelated. Considering one aspect alone can be misleading and lead to an unexpected outcome. To make the situation even more complicated, the relative importance of each criteria may vary in different situations and be applied differently in the case of different individuals.

4. Project Methodology

The system that was created references two databases. The first is of staff specialisms and experience, and the second holds the requirements of the work to be done. Heurisitic rules are then applied to these and the result is a list of names of staff suitable for, and capable of, the many tasks to be done. The databases are dynamic and are constantly updated so that the system gains experience through increasing knowledge and can act as an advisor who knows each staff member well by their experience, specialism and current workload.

The technology of creating KNOWLEDGE from the databases of FACTS involves finding out how humans reason when working in this particular problem domain. Easily formalized or algorithmic solutions do not fit in to this type of problem but so we apply heuristic methods which allow for frequent backtracking. This searching strategy makes guesses at various points during problem solving and returns to a previous decision point to make another choice when the previous guess leads to an unacceptable result. During a period of discussion and probing, the knowledge engineer and the human expert define the problem to be attacked, discover the basic concepts involved and develop the rules governing them. Since it takes experimentation and reiteration to achieve high performance the development of an expert system will be by evolution. Once a set of rules are identified the knowledge engineer returns to the knowledge acquisition stage to further refine them and to acquire more knowledge. Various test cases can be used to determine weaknesses in the knowledge base and inference structure. The development of an expert system involves almost constant revision, refining the implemented system, adjusting the rules and control structures, reformulating concepts, enriching the knowledge base and so on until the expected behaviour is obtained.

Before decision criteria and heuristic rules could be identified for assigning staff to tasks it was necessary to have a general understanding of the problem domain to identify current practice users expectation from the expert system and interviews were conducted with those involved in the assignment process. It was apparent that although major concerns and emphases of various people may vary, still, a common set of criteria could be identified.

For staff we included:

- general skills (e.g. graphics, systems methodologies)
- experience both within our institution and before (stated in terms of subject areas)
- an individual's self-development plan (in terms of areas he wished to pursue both in teaching and research)
- desirable workload (measured roughly in hours/week)
- staff grade

For jobs to be allocated we included:

- subject area
- amount of work involved and when it had to be done
- job history (who has done this or similar before)

The rules of thumb which were identified were that assignments should be based primarily on:

- each individual's stated preference
- matching an individual's general skills and experience against needs of the task
- an individual's available time

If a task had not been specifically requested by someone then we also considered:

- who has done similar work before
- who has the necessary skills

In case of a conflict, we determined priority by

- amount of expertise available to do this task. (This means that if the job requires special, relatively scarce, skills then it should be allocated first)

— the relative importance of the task (in our situation we decided that final year Honours Degree units were "more important" than first year introductory units).

Naturally the criteria which I describe here are very particular to the needs of our department. However the general applicability of the knowledge engineering process can be illustrated by such specific examples since the reader will be able to extrapolate from these examples to a problem domain with which he is familiar. Figure 1 illustrates the criteria to be included into the expert system and their relationships.

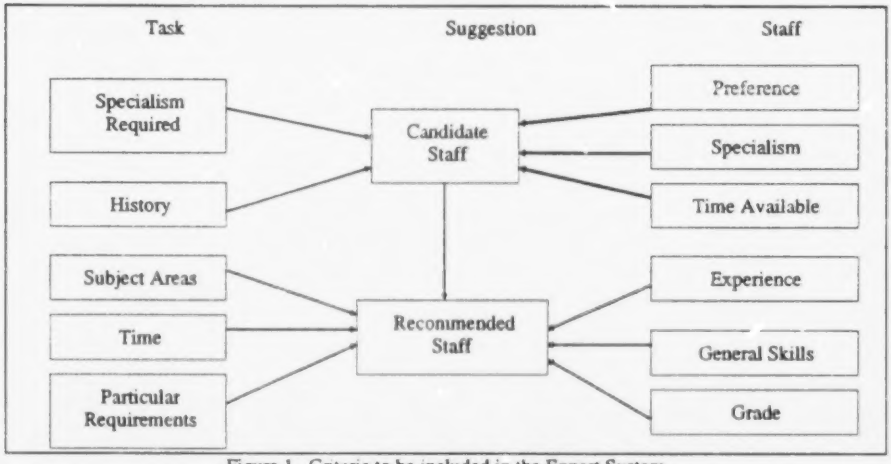

Figure 1. Criteria to be included in the Expert System

Production rules are now needed to map the required characteristics of each task on to some desirable characterisics of individual staff. More experienced staff should get more advanced tasks, less experienced staff get introductory or "medium level of difficulty" tasks, if possible people's individual preferences should be followed and so on. Figure 2 illustrates the overall theme of the inference prototyped.

Staff suggested are inferenced by a set of "staff rules" considering staff preferences, specialisms, general skills and grade. The recommended specialism, general skills and grade of staff for a particular task are in turn derived from the "heuristic rules" that match them to the task's requirements. The task's requirements are obtained by the "unit rules" using the information in the database.

5. Implementation

Our expert system was developed using a shell. This provided a domain-independent inference framework to allow the knowledge engineer to easily build, use and maintain the knowledge base of the expert system. It saves a lot of development effort but had some drawbacks which will be discussed later. Since the database which contained our knowledge was highly volatile we created our database using dBaseIII which also made it easy to update. The initial prototype was evaluated

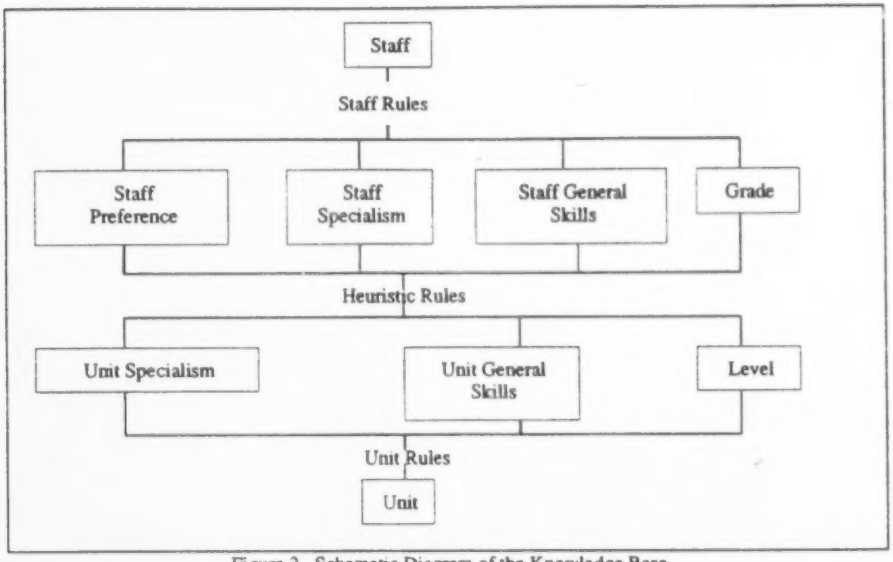

Figure 2. Schematic Diagram of the Knowledge Base

using work assignments from the previous two years to test its adequacy and reliability. Gradually the rules were refined after recommendations from the human experts who had to do a lot of work in studying lists generated by our "other" expert and discussing with the knowledge engineer why they did or did not agree with the advice given by the system.

Since this was an experimental project we decided to use available hardware and settled for an IBM PC with 640K bytes of main memory, a 10MB hard disk and two floppy disk drives. A large number of expert system languages and tools are available for this machine and since we were building a prototype our knowledge database would be easily contained on a PC. Our final software selection was

- M1 Expert System Shell Version 2.0 for building the knowledge base
- dBaseIII Version 1.10 for building and maintaining the database of staff and job information
- Microsoft C Version 4.0 and DBCIII Version 1.0 for implementing the knowledge base/database interface

The most critical consideration for software selection is really the choice of tool for building the knowledge base since it will affect the performance of the system. The choice of software for implementing the database manipulation subsystem and the interface are usually dictated by the choice of the expert system development tool. The resource allocation of staff to tasks can be compared to a problem handled in the form of a "prescription". The system will "prescribe" the most suitable person on the basis of the "symptoms" of the tasks stored in the knowledge base. M1 is designed to handle problems of this description and has been thoroughly tested against a library of sample knowledge systems. In M1 facts are represented as attribute-value pairs to which certainty

factors are attached. Heuristic knowledge is represented by production rules and a rule tests the value of an expression and thereby concludes values for another expression. The degree of confidence in information which M1 concludes is measured by a certainty factor between -100 (absolutely false) and 100 (absolutely true) with 0 meaning that no conclusion can be reached (neutral). The M1 inference engine is a backchainer. The order of clauses in a rule influence how backchaining will proceed. This allows the program to execute without a well defined algorithm since it makes guesses at various points during problem solving and returns to a previous decision point to make another choice when the guess leads to an unacceptable result. The user interface in M1 is not very friendly and is oriented to the use of computer people rather than the general user. A panel mode allows you to trace the events through a set of windows allowing display of rules under consideration, any intermediate conclusions, and options. M1 allowed us to develop a system fairly quickly since our problem matched this tool. It is easy to use given a certain amount of computer expertise and meant less time was spent on development in the first prototype which primarily aimed at exploring the effectiveness of the heuristics. Once jobs have been allocated to people we need to keep records of these facts and we designed this to be implemented as a relational database using dBaseIII since M1 can access the data through a set of C Language routines. In addition, the "flat file" design of the relational database is simple but powerful enough for this project. Indexes can be built up in non-key variables to speed up navigation of the database files during the search process of the expert system consultation.

A stand-alone subsystem was developed in dBaseIII to maintain the databases. Reports and inquiries are also available for listing various information such as details of all tasks still unallocated, lists of staff with some spare time available and so on. The "C" interface previously mentioned performs several functions. Whenever the database is required during the consultation the external interface routine retrieves the required information from the database and passes it back to the knowledge base to continue the consultation. Similarly when information in the database is updated by the knowledge base the interface routine then updates the rquired information in the database and passes a status flag back to indicate whether the process has been successful or not.

6. Evaluation

In evaluating the performance of an expert system we can look at various aspects of it such as the quality of the system's performance, decisions and advice. Reliable advice was essential for the resource allocation prototype we were building. However, the mechanisms for deciding whether advice is "adequate" may be difficult to define since the human expertise being implemented tended to be highly judgmental and non-standardised. Because of the nature of the current project we were also less concerned with the correctness of the reasoning techniques used since this tends to be more appropriate for highly technical problem domains.

The quality of the human-computer interaction is important in gaining user acceptance by end users but was also inhibited by the nature of M1 itself (for example the use of capital letters is frequently impossibe within M1). Relevant issues such as the appropriate use of words used in questions and responses etc. were addressed in the stages of prototype development but many of these concerns are equally true in any computer system and are not the particular preserve of expert systems alone.

The efficiency of the system in the decision making process in the user environment is another important evaluation consideration. It became apparent that our system was at the limit of the machine on which we were running it in terms of how long the user had to wait. A review of various aspects of the implementation could possibly allow better utilisation of computer power resulting in a better response time.

The human expert's decisions are (in all honesty) highly subjective and intuitive. It is therefore very difficult even for the expert to explicitly state the rules he applies in a step-by-step manner. It quickly became apparent that the rules of thumb extracted from the experts are often applied simultaneously and their relative importance is adjusted in different situations. We then decided that it was more reasonable to evaluate the system's performance, not by the way it simulated the human expert's reasoning process, but by its perfomance in comparison with the human. It is also difficult in the testing stage to tell whether the list produced by the prototype is "satisfactory" and "correct". Two different human experts may come up with different choice of staff for a particular job, both of whom could justify their selection as "satisfactory" and "correct".

Because this project was implemented by a student it was therefore also subject to close supervision at every stage of development and so the question of when to evaluate the system did not really arise. Evaluation was continual, beginning when the system started, extending in an informal fashion through the early stages of development and finally becoming increasingly formal as the developing expert system began to achieve real-world implementation. We consciously planned for the implementation of a Mark I version of the protoype as early as possible to demonstrate that the selected performance task was feasible, that the representation of knowledge was appropriate for the task domain and that by utilising knowledge engineering techniques that the prototype system could be built that performed reasonably well on some subtasks of the domain. This involved an informal evaluation simply showing that a few special cases could be handled by the prototype system. Further refinements followed through feedback from users. Due to the evolutionary nature of the development process evaluation is intrinsic. This project proved to be an interesting one. From the managers' point of view it made one pause to think about the way we work, to analyse one's own thought processes and perhaps to realise we are not quite such rational beings as we had imagined ourselves to be. The prototype finally developed provides reasonably accurate information about matching people with jobs to be done and has therefore demonstrated the effective use of expert systems technology in solving problems in a real world situation.

An added advantage of this type of knowledge base is that it can be used to identify relative strengths and weaknesses in an organisation. Areas of endeavour in which only one or two staff are experienced can be highlighted easily and staff recruitment can therefore be more focussed or alternatively staff training and development programmes can be instituted to address the weak areas.

Other expert systems are characterised by their ability to continue their inference processes with vague or uncertain information but this system relies on accurate information held in the database to make its decisions. For this reason it is arguable that it could just as easily have been developed through a conventional programming approach. However, apart from vastly increased development time this would also have resulted in the heuristic rules for staff optimization being embedded in the program code making it both difficult to change and to check.

Many possible extensions of this project now suggest themselves. It would be useful for example to look at an individual's overall work assignments from a global viewpoint by including heuristics for the inter-relationships between the different tasks assigned to him.

7. Conclusion

This was a valuable learning experience for those involved in this project. It explored knowledge engineering concepts and provided experience of the potential of the heuristic approach in solving non-deterministic problems. The acquisition of knowledge is one of the most difficult tasks of the knowledge engineer. The most effective ways of acquiring knowledge such as interviews are time-consuming, and in a business organisation, costly. Further problems arise when the acquired knowledge is converted to rule-form since such knowledge is frequently incomplete.

Acknowledgement

The work described in this paper was done by Ms Wong Mei Po as a final year project during her course of study for her degree B.A.(Hons) Computing Studies. The idea for the project came from Dr. K.B.C.Saxena of the Computing Studies Department and the author was the project supervisor. Mabel Wong applied herself extremely well to this project and graduated in 1987 with a well-deserved Upper Second Class Honours Degree.

References

[BUCH84] Buchanan B G, Shortliffe H, **Rule-Based Expert Systems: the MYCIN Experiments of the Stanford Heuristic Programming Project**, Ed., Addison Wesley, 1984

[HAY83] Hayes-Roth F, Waterman D A, Lenat D B, **Building Expert Systems"' Ed., Addison Wesley, 1983**

[HAY85] Hayes-Roth F, **Rule-Based Systems**, Communications of ACM, Sept. 1985

[NIL80] Nilsson N J, **Principles of Artificial Intelligence**, Tioga, 1980

[M1RM] Teknowledge Inc., **M1 Reference Manual**, 1986

[M1SK] Teknowledge Inc., **M1 Sample Knowledge Systems**, 1986

[CLCK84] Clocksin W F, Mellish C S, **Programming in PROLOG**, Springer-Verlag, 1984

[DBSRM] Ashton Tate, **dBaseIII for your 16 bit PC**, 1984

[PCRM] Texas Instruments, **Personal Consultant Reference Manual**, 1984

[DBCUM] Lattice Inc., **dBCIII User's Manual**, 1984